Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 11, No. 2, 2019 Article ID IJIM-0986, 14 pages
Research Article

Science and Research Branch (IAU)

Hybrid PSOS Algorithm For Continuous Optimization

A. Jafarian *†, B. Farnad ‡

Received Date: 2018-03-30 Revised Date: 2018-11-02 Accepted Date: 2019-03-11

Abstract

Particle swarm optimization (PSO) is one of the practical metaheuristic algorithms which is applied for numerical global optimization. It benefits from the nature inspired swarm intelligence, but it suffers from a local optima problem. Recently, another nature inspired metaheuristic called Symbiotic Organisms Search (SOS) is proposed, which doesn't have any parameters to set at start. In this paper, the PSO and SOS algorithms are combined to produce a new hybrid metaheuristic algorithm for the global optimization problem, called PSOS. In this algorithm, a minimum number of the parameters are applied which prevent the trapping in local solutions and increase the success rate, and also the SOS interaction phases are modified. The proposed algorithm consists of the PSO and the SOS phases. The PSO phase gets the experiences for each appropriate solution and checks the neighbors for a better solution, and the SOS phase benefits from the gained experiences and performs symbiotic interaction update phases. Extensive experimental results showed that the PSOS outperforms both the PSO and SOS algorithms in terms of the convergence and success rates.

Keywords : PSO; SOS; Meta-Heuristic Optimization; Hybrid Algorithm

1 Introduction

Optimization is the process of finding the best solution. The best solution means that there could be more than one solution for a problem, while only one of them is the best. Based on the optimization method, the optimization algorithms are categorized into exact algorithms and approximate algorithms. Exact algorithms are mainly applied to find the absolute solution for the problems, but they are not effective in solving the hard optimization problems and may increase the time of optimization exponentially [1]. However, approximate algorithms are the best choice for solving hard optimization problems and can find the best solution in a minimum execution time.

Approximate algorithms themselves can be classified into Heuristic algorithms and Meta-Heuristic algorithms. However, heuristic algorithms suffer from the local solutions, but meta-heuristic algorithms are aimed to avoid the local solutions which causes their popularity and widespread usage in various applications [2]. Metaheuristic algorithms combine various intelligent procedures and guide basic heuristic methods [3]. These algorithms are inspired from different things such as natural phenomena, natural selections and social behaviors and applied in solving the optimization problems. Examples of the recently metaheurtistc algorithms are HTS (heat transfer search) [4], NBA (novel bat algorithm) [5], Vortex search [6], MBA (mine blast

*Corresponding author. jafarian5594@yahoo.com, Tel: +98(914)1893041.

†Young Researchers and Elite Club, Urmia Branch, Islamic Azad University, Urmia, Iran.

‡Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran.

algorithm) [7], WCA (water cycle algorithm) [8], and SFS (stochastic fractal search) [9].

The symbiotic organism search (SOS) [10] is one of the nature inspired [11] algorithms presented to solve numerical optimization which is aimed to simulate the symbiotic interaction between the organisms, and applying them to find a better survival opportunity. Symbiotic interactions are categorized into three phases: Mutualism, Commensalism and Parasitism. These interactions are performed between a pair of organisms in the ecosystem, for example, Mutualism is like the interaction between bees and followers, and both of the organisms which participate in the Mutualism benefit from the interaction. These phases in SOS algorithm are the updating section in which the position of the organisms are changed according to the interaction. Organisms (individuals) in SOS interact with each other and only the most compatible organisms are survived and benefit from the ecosystem. At the end, the fittest organism to the ecosystem is selected as the solution to the problem.

The PSO [12] is the other metahearuistic algorithm which has been utilized in the optimization of many problems. This algorithm uses the strategy of birds and folks in migration for finding better solutions. Individuals in the PSO are called as particles and each particle has velocity in the searching space. Particles are distributed randomly in the searching space and positions of the particles are changed based on the velocity which has been calculated. These particles tend to move toward the best positions which causes to seek a better position and find the best.

One of the deficiencies that can be specified for PSO is that it often falls to the local minimum quickly, missing better opportunities when facing multimodal functions [13]. Researchers have tried to change the original version of the PSO for boosting its efficiency in finding solutions. Generally the deficiencies of the PSO and other metaheuritic optimization algorithms can be solved by two methods: first, improving the algorithm and designing the new version. Many variants of the PSO has been proposed in recent years such as: modified particle swarm optimizer (GPSO) [14], Dynamic multi-swarm particle swarm optimizer (DMS-PSO) [15], adaptive particle swarm optimizer (APSO) [16], fully informed particle swarm (FIPS) [17], comprehensive learning particle swarm optimizer (CLPSO) [18]. Second, PSO may be combined with other more efficient optimization algorithms to produce hybrid optimization algorithms. Examples of such hybrid algorithms are HP-CRO [19], ICA-PSO [20] and CS/PSO [21]. In ICA-PSO, exploration ability has been boosted by combining imperialist competitive algorithm (ICA) [22]. HP-CRO is the algorithm based on the chemical reaction optimization (CRO) [23] and CS/PSO is a combination of cuckoo search (CS) [24]. The main purpose from combing the algorithms is covering each others deficiency and boosting the problem solving ability and decreasing the number of function evaluation (NFE). Thus, the resulted hybrid algorithm should be able to solve most of the problems efficiently and with fast convergence.

In this paper, we combine the PSO and SOS algorithms to achieve a new hybrid metaheuristic algorithm for the global optimization. Combination of these two algorithms is aimed to resolve some problems which cannot be solved by the SOS and the PSO. The main reason for this improvement is originated from the modifications which have been performed in the SOS part of the proposed algorithm. These modifications cause preventing from trapping in the local solutions and increase in the success rate. Since the SOS algorithm doesn't have any parameters, the PSOS applies minimum number of the parameters and only uses the PSO parameters. In this hybrid algorithm, the PSO has the role of gaining experiences and selecting the best from them to use in the SOS interaction phases, which helps in fast convergence.

The rest of the paper is organized as follows: Section 2 describes the generic form of the optimization problems, Section 3 illustrates the PSO and SOS algorithms, and section 4 discusses the PSOS algorithm, its parameters and boundary control. Finally, the last section presents the concluding remarks.

1.1 Optimizatoin problems

Optimization problems have been inspired from the real world problems [25], the problem with more than one objective function called as multi-objective (m>1). The main purpose in the optimization is to find the global minimum or maximum. The function may have more than one minimum or maximum which is called as the local, but only one of them is the global maximum or minimum. The point x* is the global minimum if f(x*)≤f(x) for all the x in the searching space S. Optimization problem may consist of one or more mathematical functions which need to be optimized. The general form of the optimization problem is indicated in Eq. (1.1).

$$MinF(f_1(x), ..., f_m(x)), \quad x = (x_1, ..., x_n) \in S. \tag{1.1}$$

Where n is the decision variables, m is the number of objectives, x is decision vector and S is searching space. If the problem has one objective function (m=1), then it should be indicated as Eq. (1.2).

$$Minf(x), \quad x = (x_1, ..., x_n) \in S. \tag{1.2}$$

2 PSO and SOS

2.1 Particle Swarm Optimization (PSO)

The PSO is the one of the metaheuristic algorithms which is originated from the nature. This algorithm was introduced by Kenndy and Eberahart in 1995 [12]. The PSO is originated from the birds and folks migration behavior, living in small and large numbers of groups. The birds use a method for finding food and migration, which has been used in this algorithm. In this method, only the birds know their distance from food, but they don't know the location of the food thus, following the other neighboring birds is the best way for surviving.

The PSO consist of elements with the name of particles which is a probable solution in the searching space. The main steps in the PSO algorithm are as follow: first, particles are distributed randomly in the searching area and PSO starts the process with these particles. In this searching process, particles only follow the one which is nearer to the goal and has better fitness value. Each particle has a velocity which is represented by V_i and calculated by Eq. (2.3) in the D-dimensional searching space. Particles are under the effect of personal ($Pbest_i^t$) and swarm experiences ($Gbest_i^t$) and the position is updated by Eq. (2.4).

$$V_i^{t+1} = w.V_i^t + c_1.r_1.(Pbest_i^t - X_i^t) + c_2.r_2.(Gbest_i^t - X_i^t) \tag{2.3}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{2.4}$$

In Eq. (2.3) and (2.4), X_i represents the *ith* particle of the population, c_1 and c_2 are the learning coefficients, r_1 and r_2 are random values between [0 1], w is the inertia weight, and V_i is the *ith* member of particles velocity. $Pbest_i^t$ and $Gbest^t$ are the personal best and generation best.

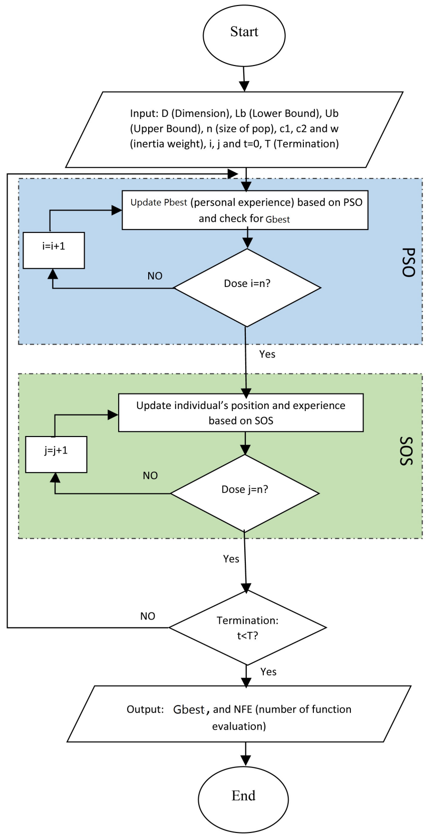

Figure 1: Flowchart of the PSOS algorithm.

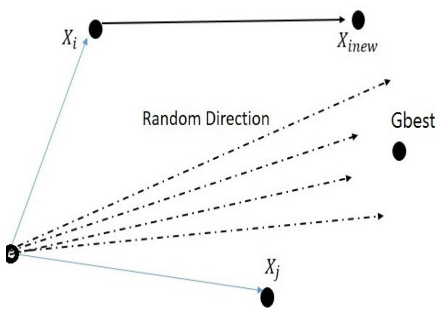

Figure 2: Modified Commensalism effects on updating position.

PSO algorithm:

Initialize locations X_i and velocity V_i of n particles.
Find *Gbest* from $\min\{f(X_1), ..., f(X_n)\}$ (at t = 0)
While (criterion)
  for i=1,2,...,n **do**
    Generate new velocity ${V_i}^{t+1}$ using Eq. (2.3).
    Calculate new locations ${X_i}^{t+1} = {X_i}^{t} + {V_i}^{t+1}$
    Evaluate objective functions at new locations ${X_i}^{t+1}$
    If ${X_i}^{t+1}$ is better than $Pbest_i^t$ **then**
    Set ${X_i}^{t+1}$ to be $Pbest_i^t$
     end if
  end for
  Find the Generation best $Gbest^t$ from particles $Pbest^t$
   iter = iter + 1 (pseudo time or iteration counter)
end while
Output the final result *Gbest*

Table 1: Unimodal test functions (D: dimensions)

Function	D	Range	Min	Formulation
F1(Beale)	2	[-4.5,4.5]	0	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2$ $+(2.625 - x_1 + x_1x_2^3)^2$
F2(Easom)	2	[-100,100]	-1	$f(x) = -\cos(x_1)\cos(x_2)\, exp(-(x_1-\pi)^2 - (x_2-\pi)^2)$
F3(Matyas)	2	[-10,10]	0	$f(x) = 0.26\,(x_1^2 + x_2^2) - 0.48x_1x_2$
F4(Colville)	4	[-10,10]	0	$f(x) = 100(x_1^2 - x_2)^2 + (x_1-1)^2 + (x_3-1)^2$ $+90(x_3^2 - x_4)^2 + 10.1(x_2-1)^2 + (x_4-1)^2$ $+19.8\,(x_2-1)\,(x_4-1)$
F5(Zakharov)	10	[-5,10]	0	$f(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$
F6(Schwefel 2.22)	30	[-10,10]	0	$f(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$
F7(Schewefel 1.2)	30	[-100,100]	0	$f(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$
F8(Dixon-price)	30	[-10,10]	0	$f(x) = (x_1-1)^2 + \sum_{i=2}^{D} i\,(2x_i^2 - x_{i-1})^2$
F9(Step)	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^{D} (x_i + 0\cdot 5)^2$
F10(Sphere)	30	[-100,100]	0	$f(x) = \sum_{i=1}^{D} x_i^2$
F11(SumSquares)	30	[-10,10]	0	$f(x) = \sum_{i=1}^{D} ix_i^2$
F12(Quartic)	30	[-1.28,1.28]	0	$f(x) = \sum_{i=1}^{D} ix_i^4 + Rand$

2.2 Symbiotic Organisms Search (SOS)

The SOS is introduced by Cheng and Prayogo in 2014 [10] and is the one of nature inspired meta-heursitic algorithms which applies the symbiotic interactions to survive organism in the ecosystem. The SOS tries to find the best survival opportunity by using the symbiotic behaviors

Table 2: Multimodal test functions (D: dimensions)

Function	D	Range	Min	Formulation
F13(Schaffer)	2	[-100,100]	0	$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{(1+0.001(x_1^2+x_2^2))^2}$
F14(6 H Camel)	2	[-5,5]	-1.03163	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
F15(Bohachevsky2)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)(4\pi x_2) + 0.3$
F16(Bohachevsky3)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$
F17(Shubert)	2	[-10,10]	-186.73	$f(x) = (\sum_{i=1}^{5} i\cos(i+1)x_1 + i)$ $(\sum_{i=1}^{5} i\cos((i+1)x_2 + i))$
F18(Rosenbrock)	30	[-30,30]	0	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
F19(Griewank)	30	[-600,600]	0	$f(x) = \frac{1}{4000}\left(\sum_{i=1}^{D}(x_i - 100)^2\right)$ $-\left(\prod_{i=1}^{D}\cos\left(\frac{x_i-100}{\sqrt{i}}\right)\right) + 1$
F20(Ackley)	30	[-32,32]	0	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$
F21(Bohachevsky1)	2	[-100,100]	0	$f(x) = x_1^2 + 2x_2^2 - 0\cdot 3\cos(3\pi x_1)$ $-0.4\cos(4\pi x_2) + 0.7$
F22(Booth)	2	[-10,10]	0	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
F23(Michalewicz2)	2	[0,π]	-1.8013	$f(x) = -\sum_{i=1}^{D} sin(x_i)(\sin(ix_i^2/\pi))^{20}$
F24(Michalewicz5)	5	[0,π]	-4.6877	$f(x) = -\sum_{i=1}^{D} sin(x_i)(\sin(ix_i^2/\pi))^{20}$
F25(Michalewicz10)	10	[0,π]	-9.6602	$f(x) = -\sum_{i=1}^{D} sin(x_i)(\sin(ix_i^2/\pi))^{20}$
F26(Rastrigin)	30	[-5.12,5.12]	0	$f(x) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$

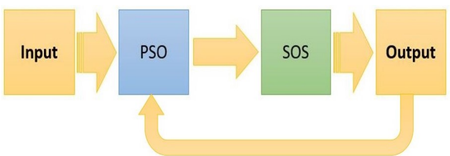

Figure 3: Schematic view of the PSOS algorithm.

which are common between the organisms. In this algorithm, organisms have the role of points in the searching space and each organism represents a possible solution to the problem. The adaption to the intended goal has been measured by the fitness for each organism. the same as the real GA [26] and continuous PSO, the SOS is likewise designed for the continuous search space. Most metaheuaristic algorithms need parameters to be tuned at the start, which has some effects on finding the proper solution, but the SOS has no parameters to be adjusted. This advantage causes the SOS to be more preferable in combination with the other algorithms. For increasing the degree of adaption, it needs to update the organism position in each generation. Updating the position in each generation is performed by updating operators which simulate the biological interaction model Mutualism/Commensalism/Parasitism. These are the most common symbiotic interactions in the nature. The SOS consist of three updating phases first phase is the Mutualism which both of

Table 3: PSOS comparison with GA, DE, PSO, BA, PBA and SOS (unimodal function set), bold values represent the best.

Function	GA	DE	PSO	BA	PBA	SOS	PSOS
(F1)Mean	**0**	**0**	**0**	1.88E-05	**0**	**0**	**0**
StdDev	**0**	**0**	**0**	1.94E-05	**0**	**0**	**0**
Rank	1	1	1	2	1	1	1
(F2)Mean	**-1**	**-1**	**-1**	-0.99994	**-1**	**-1**	**-1**
StdDev	**0**	**0**	**0**	4.50E-05	**0**	**0**	**0**
Rank	1	1	1	2	1	1	1
(F3)Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Rank	1	1	1	1	1	1	1
(F4)Mean	0.01494	0.04091	**0**	1.11760	**0**	**0**	**0**
StdDev	0.00736	0.08198	**0**	0.46623	**0**	**0**	**0**
Rank	2	3	1	4	1	1	1
(F5)Mean	0.01336	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	0.00453	**0**	**0**	**0**	**0**	**0**	**0**
Rank	2	1	1	1	1	1	1
(F6)Mean	11.0214	**0**	**0**	**0**	7.59E-10	**0**	**0**
StdDev	1.38686	**0**	**0**	**0**	7.10E-10	**0**	**0**
Rank	3	1	1	1	2	1	1
(F7)Mean	7.40E+03	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	1.14E+03	**0**	**0**	**0**	**0**	**0**	**0**
Rankv	2	1	1	1	1	1	1
(F8)Mean	1.22E+03	0.66667	0.66667	0.66667	0.66667	0.66667	**0.003727**
StdDev	2.66E+02	E-9	E-8	1.16E-09	5.65E-10	0	**0.002166**
Rank	3	2	2	2	2	2	1
(F9)Mean	1.17E+03	**0**	**0**	5.12370	**0**	**0**	**0**
StdDev	76.56145	**0**	**0**	0.39209	**0**	**0**	**0**
Rank	3	1	1	2	1	1	1
(F10)Mean	1.11E+03	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	74.21447	**0**	**0**	**0**	**0**	**0**	**0**
Rank	2	1	1	1	1	1	1
(F11)Mean	1.48E+02	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	12.40929	**0**	**0**	**0**	**0**	**0**	**0**
Rank	2	1	1	1	1	1	1
(F12)Mean	0.18070	0.00136	0.00116	**1.72E-06**	0.00678	9.13E-05	2.45E-05
StdDev	0.02712	0.00042	0.00028	**1.85E-06**	0.00133	3.71E-05	1.96E-05
Rank	7	5	4	1	6	3	2
Average rank	2.41	1.58	1.33	1.58	1.58	1.25	1.08
Overall rank	5	4	3	4	4	2	1

the organisms benefit, similar to the relationship between bees and flowers. In second phase If one benefits and the other unaffected, it is called as Commensalism, remora fish and sharks have this kind of relationship where remora receives some benefits, while shark is unaffected from this relationship (neither benefits nor suffers). The relationship that one benefits and the other is harmed, is called Parasitism and this is the last phase. This relationship is seen in anopheles mosquito relation with humans where mosquito benefits and human is harmed and may die. In each generation of the SOS, updating the position of an organism occurs if the specific relationship (Mutualism/Commensalism/Parasitism) causes a better fitness value for organisms i or j.

In each iteration, organisms position is updated as follows:

Table 4: PSOS comparison with GA, DE, PSO, BA, PBA and SOS (multimodal function set), bold values represent the best.

Function	GA	DE	PSO	BA	PBA	SOS	PSOS
(F13)Mean	0.00424	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	0.00476	**0**	**0**	**0**	**0**	**0**	**0**
Rank	2	1	1	1	1	1	1
(F14)Mean	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**
StdDev	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Rank	1	1	1	1	1	1	1
(F15)Mean	0.06829	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	0.07822	**0**	**0**	**0**	**0**	**0**	**0**
Rank	2	1	1	1	1	1	1
(F16)Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Rank	1	1	1	1	1	1	1
(F17)Mean	**-186.73**	**-186.73**	**-186.73**	**-186.73**	**-186.73**	**-186.73**	**-186.73**
StdDev	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Rank	1	1	1	1	1	1	1
(F18)Mean	1.96E+05	18.20394	15.088617	28.834	4.2831	1.04E-07	**0**
StdDev	3.85E+04	5.03619	24.170196	0.10597	5.7877	2.95E-07	**0**
Rank	7	5	4	6	3	2	1
(F19)Mean	10.63346	0.00148	0.01739	**0**	0.00468	**0**	**0**
StdDev	1.16146	0.00296	0.02081	**0**	0.00672	**0**	**0**
Rank	5	2	4	1	3	1	1
(F20)Mean	14.67178	**0**	0.16462	**0**	3.12E-08	**0**	**0**
StdDev	0.17814	**0**	0.49387	**0**	3.98E-08	**0**	**0**
Rank	4	1	3	1	2	1	1
(F21)Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**
StdDev	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Rank	1	1	1	1	1	1	1
(F22)Mean	**0**	**0**	**0**	0.00053	**0**	**0**	**0**
StdDev	**0**	**0**	**0**	0.00074	**0**	**0**	**0**
Rank	1	1	1	2	1	1	1
(F23)Mean	**-1.8013**	**-1.8013**	-1.57287	**-1.8013**	**-1.8013**	**-1.8013**	**-1.8013**
StdDev	**0**	**0**	0.11986	**0**	**0**	**0**	**0**
Rank	1	1	2	1	1	1	1
(F24)Mean	-4.64483	-4.68348	-2.4908	**-4.6877**	**-4.6877**	**-4.6877**	**-4.6877**
StdDev	0.09785	0.01253	0.25695	**0**	**0**	**0**	**0**
Rank	3	2	4	1	1	1	1
(F25)Mean	-9.49683	-9.59115	-4.0071	**-9.6602**	**-9.6602**	-9.65982	**-9.6602**
StdDev	0.14112	0.06421	0.50263	**0**	**0**	0.00125	**0**
Rank	4	3	5	1	1	2	1
(F26)Mean	52.92259	11.71673	43.97714	**0**	**0**	**0**	**0**
StdDev	4.56486	2.53817	11.72868	**0**	**0**	**0**	**0**
Rank	4	2	3	1	1	1	1
Average rank	2.64	1.64	2.28	1.42	1.35	1.14	1
Overall rank	7	5	6	4	3	2	1

2.2.1 Mutualism phase

In this updating phase, both of the organisms benefit from the relationship and need to update their positions. Two organisms applied in this phase are indicated with X_i and X_j in which organism X_i is related to the *ith* member of the population and organism X_j is selected randomly to interact with each other. New candidate positions for the organisms i and j are computed by

SOS algorithm:

1: **Input:** objective function f, Lb (Lower bound), Ub (Upper bound) and the dimensions of the problem (D)
2: \\Initialization
3: Initial parameters n (population size)
4: Let population be the set of organism 1, 2, . . . , n
5: **for** each of organism **do**
6: Assign random real number between (Lb , Ub) to the organism position
7: Calculate fitness (cost) for assigned position
8: **end for**
9: \\iterations
10: **while** (the stopping criterion is not met) **do**
11: Identify best organism X_{best}
11: **for** each of Organism i (X_i) **do**
12: \\Apply Mutualism to X_j and X_i
13: Select one organism randomly, X_j, where $X_j \neq X_i$
14: Determine mutual relationship vector (Mutual_Vector) by Eq. (2.7)
14: Calculate X_{inew} and X_{jnew} by Eq. (2.5) and (2.6)
15: **if** X_{inew} fitter than X_i **then**
16: Update X_i position with X_{inew}
17: **end if**
18: **if** X_{jnew} fitter than X_j **then**
19: Update X_j position with X_{jnew}
20: **end if**
21: \\Apply Commensalism to X_i
22: Select one organism randomly, X_j, where $X_j \neq X_i$
22: Calculate X_{inew} by Eq. (2.8)
23: **if** X_{inew} fitter than X_i **then**
24: Update X_i position with X_{inew}
25: **end if**
26: \\Apply Parasitism to X_j
27: Select one organism randomly, X_j, where $X_j \neq X_i$
28: Create a Parasite (Parasite_Vector) from Organism X_i
29: **if** Parasite_Vector fitter than X_j **then**
30: Update X_j position with Parasite_Vector
31: **end if**
32: **end for**
33: **end** while
34: \\the final stage
35: output the minimum value found (X_{best})

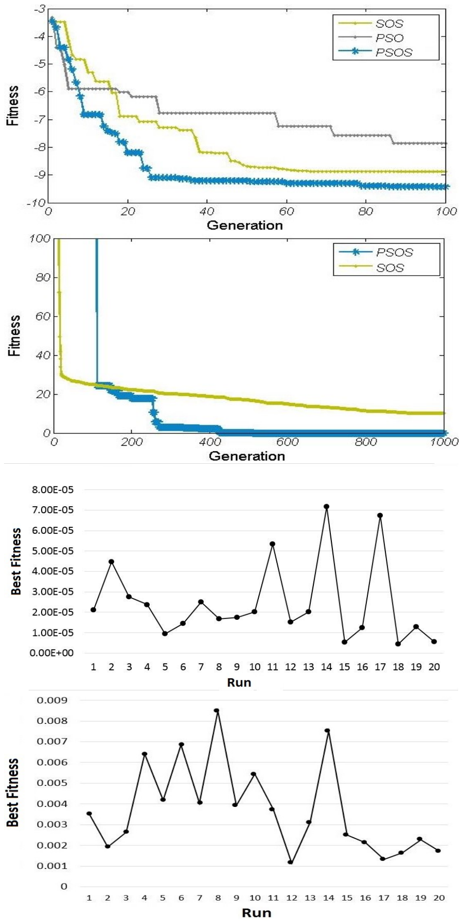

Figure 4: Convergence diagram for functions (a) F25 and (b) F18, stability diagram for functions (c) F12 and (d) F8. Stability diagram has been showed for 20 runs.

Eq. (2.5) and (2.6).

$$X_{inew} = X_i + rand(0,1) * (X_{best} - Mutual\_vector * BF_1) \quad (2.5)$$

$$X_{jnew} = X_j + rand(0,1) * (X_{best} - Mutual\_vector * BF_2) \quad (2.6)$$

$$Mutual\_vector = \frac{X_i + X_j}{2} \quad (2.7)$$

The relationship characteristic between organisms X_i and X_j is indicated by $Mutal\_vector$ in Eq. (2.7). BF_1 and BF_2 in Eq. (2.5) and (2.6) represent the level of benefits to each organism from the interaction being a benefit factor which is randomly chosen, either 1 or 2. X_{best} is the best solution that has ever been found.

2.2.2 Commensalism phase

X_j is selected randomly from the searching space and interacts with organism X_i, but this time, only X_i benefits from this relationship and will be updated, and nothing happens to the organism X_j. Update equation for this phase has been showed in Eq. (2.8) .

$$X_{inew} = X_i + rand(-1,1) * (X_{best} - X_j) \quad (2.8)$$

2.2.3 Parasitism phase

In this relationship, only the organism which benefits from the relation, will be updated, and the other will be killed or harmed and will need to be replaced with the other. Organism X_j is selected randomly and serves as a victim for the parasite vector. Parasite vector is created in the search space by duplicating X_i, then modifying the randomly selected dimension using random numbers. If the parasite vector is more adopted than the selected X_j, then it kills X_j and takes its place; otherwise, X_j will have inviolability from the parasite and can live longer than that. The exploitation and exploration are two main abilities for finding a solution in the metaheuristic algorithms. Our proposed algorithm is aimed to balance the exploitation and exploration by combining the SOS and PSO algorithms. Generally, the SOS algorithm doesn't have learning strategies and swarms intelligence which could be provided by the PSO. In the SOS, personal experiences don't have any effect on the process of finding a solution and each organism is under the effect of symbiotic interactions by using Gbest. The PSO tries to move the population to the best by using the swarm intelligence and personal experiences. Proposed hybrid algorithm PSOS has been created By using the swarm intelligence of the PSO and combining the components of these two algorithms. Flowchart of this algorithm is presented in Fig. 1. In this hybrid algorithm, neighbors of every possible solution is checked by the PSO update equation, and personal experiences (Pbest) have also effects on finding a solution. For a better performance of the PSOS, the commensalism phase update equation is also changed by using a R which is a random integer number between 1 and D (dimension). The alteration in the commensalism leads into checking the neighbors greatly than before.

The proposed hybrid algorithm begins by

searching the initial population (pop) and initial size (n). Each possible solution is consisted of 5 variables: velocity, position, cost, best personal experience Pbest (which itself consists of two fields position and cost). After initializing, the main iteration of algorithm starts with the PSO. In this phase, each element of the population is checked by the PSO for finding better experiences. If the movement causes a greater position for the organism, updating best experience (Pbest) is applied for it, otherwise nothing changes. If the personal best experience (Pbest) for organisms is better than Gbest, then Gbest is also updated. Since the personal best experiences have been gained for each organism and best solution (Gbest) has been chosen from them, Gbest is used for having a better symbiotic interaction (Mutualism, Commensalism, Parasitism). In the SOS phase, a process is performed being the same as the original SOS, but this time only uses a different equation in commensalism phase of this algorithm. The process in this phase is like this, if the specific interaction led into a greater place, the organism position (X_i) and personal best experience (Pbest) are updated by the SOS interaction phases, otherwise nothing changes. Fig. 3 demonstrates the schematic view of the PSOS. The iteration best is Gbest and the algorithm stops if the ending conditions are acceptable, otherwise, the next iteration starts generating solutions and passing the another generation.

2.3 Modified Commensalism Phase

The variation in commensalism performed by R, which is a random dimension from {1,2,...,D}, has been showed in Eq. (9). For example in problem with 5 dimensional variables, Random dimension could be R = 1,2,3,4 or 5. Choosing random dimension causes movement toward the best in different directions and seek the neighbor areas greatly than previous mode of commensalism update equation Fig. 2. The advantage of this alteration is a faster convergence and avoidance of trapping in local solutions.

$$X_{inew} = X_i + rand(-1,1) * (Gbest(R) - X_j) \tag{2.9}$$

For further explanation of How random dimension (R) working in commensalism equation, Following numerical example has been proposed. Suppose that:

X_i=[0.4 0.2 0.3 0.7],
X_j =[0.3 0.3 0.2 0.4],
X_{best}=[0.2 0.3 0.5 0.6].
rand(-1,1)=0.3
Select R from $\{1, 2, 3, 4\}$ randomly. For example R=4 then $X_{best}(R)$= 0.6
X_{inew} calculated as follow:

$$\begin{aligned}
&X_{inew} = [0.4\ \ 0.2\ \ 0.3\ \ 0.7] \\
&+ 0.3 * (0.6 - [0.3\ \ 0.3\ \ 0.2\ \ 0.4]) = \\
&[0.4\ \ 0.2\ \ 0.3\ \ 0.7] + 0.3 * [0.3\ \ 0.3\ \ 0.4\ \ 0.2] = \\
&[0.4\ \ 0.2\ \ 0.3\ \ 0.7] + [0.09\ \ 0.09\ \ 0.12\ \ 0.06] = \\
&[0.49\ \ 0.29\ \ 0.42\ \ 0.76]
\end{aligned}$$

2.4 Parameter adjustments and boundary control

To get a better result in solving various problems it is needed to adjust the parameters of the algorithms properly and control the boundary when the algorithm finds a new solution [27]. The PSOS needs boundary control for a probable solution X_i because it needs a solution to be in the searching space, which is a boundary between $[Lb\ \ Ub]$ (Lb is the lower bound and Ub is the upper bound of the searching space). There are many different methods for the boundary control, in this article, it is specified in Eq. (10).

$$a = Maximum(X, Lb), b = Minimum(a, Ub) \tag{2.10}$$

Where Minimum and Maximum are the functions that select minimum and maximum among the input pairs, X is the input and b is the output which have been controlled in the boundary range $[Lb\ \ Ub]$. The PSOS parameters are n (population size), w (inertia weight), c_1 (cognitive/local weight), c_2 (social/global weight) which need to be adjusted properly for solving benchmark functions.

3 Benchmark test functions

In this article, a set of benchmark functions are used for a complete evaluation of the proposed hybrid algorithm. These benchmark functions are unimodal and multimodal functions which have

PSOS algorithm:

```
Input: objective function f, constraints and the dimensions of the problem (D)
\\Initialization
Initial parameters n (population size), w, c1, c2
Let population be the set of organism 1, 2, . . . , n
for each of organism do
  Assign random real number between (varmin , varmax) to the organism position
  Calculate fitness (cost) for assigned position and set velocity=0
  Set calculated position and cost as best experience (Pbest) position and cost too.
end for
while (the stopping criterion is not met) do
  for each of Organism i (Xi) do
    Calculate Xinew for Xi with PSO update operator Eq. (2.3)& (2.4)
    if Xinew fitter than Xi experience (Pbest) then
      Update Xi experience (Pbest) with Xinew
      if Xi experience better than Gbest then
        Update Gbest with Xi experience (Pbest)
      end if
    end if
  end for
  for each of Organism j (Xj) do
    \\Apply Mutualism to Xr and Xj
    Select one organism randomly, Xr, where Xr ≠Xj
    Determine mutual relationship vector (Mutual_Vector) by Eq. (2.7)
    Calculate Xjnew and Xrnew by Eq. (2.5) and (2.6)
    if Xjnew fitter than Xj then
      Update Xj experience and position with Xjnew
    end if
    if Xrnew fitter than Xr then
      Update Xr experience and position with Xrnew
    end if
    \\Apply Commensalism to Xj
    Select one organism randomly, Xr, where Xr ≠Xj
    Calculate Xjnew by Eq. (2.9)
    if Xjnew fitter than Xj then
      Update Xj experience and position with Xjnew
    end if
    \\Apply Parasitism to Xr
    Select one organism randomly, Xr, where Xr ≠Xj
    Create a Parasite (Parasite_Vector) from Organism Xj
    if Parasite_Vector fitter than Xr then
      Update Xr experience and position with Parasite_Vector
    end if
  end for
  decrease the inertia weight (w*wdamp wdamp is real value between (0 1))
end while
```

various dimensions such as 2, 4, 10 and 30. Tables 1 and 2 present the unimodal and multimodal benchmark functions. The PSOS has been tested by these functions and tries to find minimum solution for them. the obtained result was compared with other famous algorithms GA [28], DE [29], PSO, BA [30], PBA [31] and SOS and comparsion result has been presented in Tables 3 and 4. All conditions of the experiments by algorithms are the same and previously mentioned by Cheng and Lien who have used the population size of n=50. Parameters for the PSOS was set to the c1 and c2 =2 and w=1. The following testing setup is applied in this scheme: CPU 2.1 GHZ, Ram 8 GB and Matlab 2013 running on computer with windows 8. Stopping criteria was set to reach number of function evaluation NFE= 500,000 and the results minimum than 1E-12, reported as 0 like the other algorithms. Table 3 and 4 show the results for this experiment and It could be concluded from these tables that the PSOS can solve all the problems except Quartic (F12) and Dixon-price (F8) which have not been solved with others ,either. In these tables, if the algorithm could solves the specific function, rank 1 can be given, otherwise rank is 0. At the end of each table the average and overall for these ranks show that the PSOS has superiority in solving benchmark functions than the others. The Mean value and StdDev (standard deviation) have been calculated from 30 independent runs. For further examining of the PSOS algorithm, convergence and stability diagrams have been proposed in Fig. 4. Convergence curves in Fig. 4(a) and (b) show that the proposed hybrid method could reach the best solution faster with a minimum number of the generations. It could solve functions F25 and F18 in minimum generation than the PSO and SOS. For further evaluation, stability diagrams presented in Fig. 4(c) and (d) for 20 continuous runs of functions F12 and F8. Points in this diagram presents the best solution on a specific run. It can be concluded from the stability diagrams that the PSOS solutions are concentrated in a specific boundary for the noisy function like the Quartic function (F12).

4 Conclusion & Future Works

The SOS is a recently proposed nature inspired metaheuristic algorithm, which has shown good performance in various optimizing numerical problems. One of the practical metaheuristic algorithms which has been used most widely in the optimization is the PSO. However, the original PSO has some deficiencies in facing multimodal functions which cause trapping in the local solutions and converging to them quickly. In this paper, we have combined the SOS and PSO algorithms to produce a new hybrid algorithm for finding the global minimum. Also the PSOS applies a minimum number of the parameters which have simplified it. In the proposed algorithm, some parts of the SOS algorithm is modified to prevent trapping in the local solutions and to increase the success rate. Moreover, changes applied in the SOS Commensalism operator, have led into solving most benchmark problems with a better performance than the PSO and SOS algorithms. The proposed hybrid algorithm begins to search from the PSO algorithm and gain experiences for the SOS. The SOS algorithm uses the best experience which has been gained from the PSO and tries to find better positions. We have concluded from the experiments that the PSOS is more dominant in the optimization of the proposed functions than the other mentioned algorithms and it could score the best rank for solving benchmark problems and reach the global minimum in a limited number of function evaluations. Our future work would consist of using the proposed hybrid method for optimizing the constrained benchmark problems and engineering design problems. Also, the future researches can include a comparison of our hybrid method with the recently proposed algorithms in solving the economic dispatch problem.

References

[1] C. Grosan, A. Abraham, Intelligent systems a modern approach, *Intelligent Systems Reference Library*, Springer Berlin Heidelberg 7 (2011) 2-11.

[2] X-S. Yang, Introduction to Mathematical Optimization: From Linear Programming to Metaheuristics, *Cambridge International Science Publishing* (2008).

[3] P. Hansen, N. Mladenovic, J. Brimberg, J. Moreno-Prez, Variableneighbor-hood search,

In: Handbook of Metaheuristics, Springer, US 6 (2010) 61-86.

[4] V. K. Patel, V. J. Savsani, Heat transfer search (hts): a novel optimization algorithm, *Information Sciences* 324 (2015) 217-246.

[5] X. B. Meng, X. Gao, Y. Liu, H. Zhang, A novel bat algorithm with habitat selection and doppler effect in echoes for optimization, *Expert Systems with Applications* 42 (2015) 6350-6364.

[6] B. Doğan, T. Ölmez, A new metaheuristic for numerical function optimization: Vortex search algorithm, *Information Sciences* 293 (2015) 125-145.

[7] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm for optimization of truss structures with discrete variables, *Computers & Structures* 103 (2012) 49-63.

[8] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures* 111 (2012) 151-166.

[9] H. Salimi, Stochastic fractal search: A powerful metaheuristic algorithm, *Knowledge-Based Systems* 75 (2015) 1-18.

[10] M. Y. Cheng, D. Prayogo, Symbiotic organisms search: A new metaheuristic optimization algorithm, *Computers & Structures* 139 (2014) 98-112.

[11] X. S. Yang, Nature-Inspired Metaheuristic Algorithms: Second Edition, *Luniver Press* (2010).

[12] J. Kennedy, R. Eberhart, Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. Perth, *Australia* 11 (1995) 1942-1948.

[13] T. KAY CHEN , Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I, *Springer-Verlag Berlin Heidelberg*, 6145 (2010).

[14] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, *In: Proceedings of the IEEE Congress on Evolutionary Computation* 7 (1998) 69-73.

[15] J. J. Liang, P. N. Suganthan, Dynamic multi-swarm particle swarm optimizer, *In: Proceedings of Swarm Intelligence Symposium* 7 (2005) 124-129.

[16] Z. Zhan, J. Zhang, Y. Li, H. S. Chung, Adaptive particle swarm optimization, *IEEE Trans. Syst. Man Cybern,* Part B 39 (2009) 1362-1381.

[17] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, *IEEE Trans. Evol. Comput.* 83 (2004) 204-210.

[18] J. J. Liang, A. k. Qin, P. N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE Trans. Evol. Comput.* 103 (2006) 281-295.

[19] T. T. Nguyen, Z. Li, S. Zhang, T. K. Truong, A hybrid algorithm based on particle swarm and chemical reaction optimization, *Expert Systems with Applications* 41 (2014) 2134-2143.

[20] L. Idoumghar, N. Chrin, P. Siarry, R. Roche, A. Miraoui, Hybrid ICA-PSO algorithm for continuous optimization,*Applied Mathematics and Computation*, 219 (2013) 11149-11170.

[21] A. Ghodrati, S. Lotfi, A hybrid cs/pso algorithm for global optimization, In: J. S. Pan, S. M. Chen, N. Nguyen (Eds.), Intelligent Information and Database Systems, *Lecture Notes in Computer Science, Springer Berlin Heidelberg* 17 (2012) 89-98.

[22] E. Atashpaz-Gargari, C. Lucas, Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition, *IEEE Congress on Evolutionary Computation* 7 (2007) 4661-4666.

[23] A. Y. Lam, V. O. Li, Chemical reaction optimization: A tutorial, *Memetic Computing* 4 (2012) 3-17.

[24] X. S. Yang; S. Deb, Cuckoo search via Lvy flights. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), *IEEE Publications* 8 (2009) 210-214.

[25] S. Boyd, L. Vandenberghe, Convex Optimization, *Cambridge University Press*, 8 (2004) 129-140.

[26] L. J. Eshelman, D. J. Shaffer, Real-coded genetic algorithms and intervalschemata In D. L.Whitley (ed.), *Foundations of Genetic Algorithms* 2. San Mateo, CA: Morgan Kaufman (1993) 187-202.

[27] W. Chu, X. Gao, S. Sorooshian, Handling boundary constraints for particle swarm optimization in high-dimensional search space, *Information Sciences* 181 (2011) 4569-4581.

[28] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, *Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA* (1989).

[29] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11 (1997) 341-359.

[30] D. Pham, A. Ghanbarzadeh, E. Ko, S. Otri, S. Rahim, M. Zaidi, The bees algorithm a novel tool for complex optimisation problems, in: D. Pham, E. Eldukhri, A. Soroka (Eds.), *Intelligent Production Machines and Systems*, Elsevier Science Ltd, Oxford. 5 (2006) 454-459.

[31] M. Cheng, L. Lien, Hybrid artificial intelligencebased pba for benchmark functions and facility layout design optimization, *Journal of Computing in Civil Engineering* 26 (2012) 612-624.

Ahmad Jafarian was born in 1978 in Tehran, Iran. He received B. Sc (1997) in Applied mathematics and M.Sc. in applied mathematics from Islamic Azad University Lahi-jan Branh to Lahijan, Ferdowsi University of Mashhad respectively. He is a Assistant Prof. in the department of mathematics at Islamic Azad University, Urmia Branch, roumieh, in Iran. His current interest is in artificial intelligence, solving nonlinear problem and fuzzy mathematics.

Behnam Farnad was born in 1990 in Urmia, Iran. He Studied at National Organization for Development of Exceptional Talents (NODET) high school and received B.Sc (2008) in Computer Software engineering and M.Sc. in Computer Software engineering from Kharazmi University of Tehran, Islamic Azad University of Urmia respectively. His current interest is in artificial intelligence, solving nonlinear problem, robotics and hardware engineering.